

# WireGuard

- MTU Größe für Wireguard Verbindung berechnen
- WireGuard mit WebInterface auf Debian Server installieren

# MTU Größe für

# Wireguard Verbindung berechnen

## Einleitung

Bei der Einrichtung von einem **Wireguard Server** sollte eine **MTU-Size** mitgegeben werden. Diese gibt an, wie groß ein **VPN-Paket** mit Nutzdaten voll sein darf. In dieser Anleitung berechnen wir kurz unsere **MTU-Size**, um so die höchste **Performance** aus unserem **Wireguard Server** herauszuholen.

Wireguard verwendet eine **Standard MTU** von **1420**. Dies kann zu Problemen führen wenn dadurch das **Paket** zu groß ist und somit nicht an den **Server** oder an den **Client** übermittelt werden kann.

## Berechnung

Die Berechnung unterscheidet sich je nachdem welchen Anschluss ihr als eure Internetanbindung verwendet. Wenn es im Einzelfall nicht funktionieren sollte, prüft bitte, welche MTU-Größe ihr für euren Provider benötigt. Ich berechne die MTU-Größe beispielsweise für **DSL** mit einer **MTU** von **1492** und für **Kabel-Internet** mit einer **MTU** von **1500**.

Anhand der nachfolgenden Tabelle können die einzelnen **Byte-Größen** entnommen werden, die zur Berechnung der **MTU** benötigt werden:

Beschreibung	Byte Größe
IPv4-Verbindung	20 Bytes
IPv6-Verbindung	40 Bytes
UDP-Paket	8 Bytes
Wireguard Overhead	32 Bytes

Dementsprechend lassen sich folgende Beispiele berechnen:

<b>Beispiel 1</b>	<b>Beispiel 2</b>
-------------------	-------------------

<b>MTU DSL</b>	<b>MTU Kabel-Internet</b>
1492 MTU (Provider)	1500 MTU (Provider)
- 20 (IPv4)	- 40 (IPv6)
- 8 (UDP)	- 8 (UDP)
- 32 (Wireguard Overhead)	- 32 (Wireguard Overhead)
<b>= MTU 1432</b>	<b>= MTU 1420</b>

Die **DSL Wireguard Verbindung** verwendet im Rechenbeispiel eine Verbindung  $\frac{1}{4}$ ber **IPv4**, wohin gegen die **Kabel Wireguard Verbindung** eine Verbindung  $\frac{1}{4}$ ber **IPv6** herstellen möchte.

# WireGuard mit WebInterface auf Debian Server installieren

## Einleitung

Wenn wir unseren eigenen **WireGuard Server** betreiben möchten, wollen wir diesen vielleicht ja auch über eine **Weboberfläche administrieren**. Als Oberfläche bedienen wir uns dann an **wireguard-ui** [Link](#).

Mithilfe von **WireGuard** können wir dann ganz einfach eine **VPN-Verbindung** in unser Netzwerk herstellen und so einerseits auf unser **Heimnetzwerk** zugreifen, als auch **verschlüsseln** **Datenverkehr** nutzen.

## Vorbereitungen durchführen und WireGuard installieren

Im ersten Schritt installieren wir alle benötigten Pakete auf unserem **Debian Server**, damit wir **WireGuard** betreiben können.

```
apt update && apt install -y wireguard curl iptables tar
cd /etc/wireguard
```

Wir müssen auch sicherstellen das unsere **Firewall** Port **51820/udp** durchlässt. Und das wir ggf. unseren **WireGuard Server** als **Exposed Host** oder durch **Port NAT** unseren Server von außen erreichbar machen. Wie genau das funktioniert, bitte dem Handbuch des Herstellers der Firewall entnehmen.

Jetzt müssen wir das **IP-Forwarding** aktivieren. Dies hat den Hintergrund, dass der gesamte Verkehr auf dem **WireGuard Interface** weitergeleitet wird.

```
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
sysctl -p
```

## WireGuard UI installieren

Jetzt müssen wir folgenden Code ausführen, damit wir das **Startskript** für **WireGuard UI** erstellt wird. Damit wird der **Webserver** gestartet, und die **Datenbank** unter `/etc/wireguard/db/` abgelegt.

```
cat <<EOF > /etc/wireguard/start-wgui.sh
#!/bin/bash

cd /etc/wireguard
./wireguard-ui -bind-address 0.0.0.0:5000
EOF
chmod +x start-wgui.sh
```

Mit dem folgenden Skript wird **WireGuard UI** installiert und die entsprechenden **Dienste** angelegt. Im Anschluss wird der Download von **WireGuard UI** gestartet und der **Web Server** wird gestartet.

```
cat <<EOF > /etc/systemd/system/wgui-web.service
[Unit]
Description=WireGuard UI

[Service]
Type=simple
ExecStart=/etc/wireguard/start-wgui.sh

[Install]
WantedBy=multi-user.target
EOF

cat <<EOF > /etc/wireguard/update.sh
#!/bin/bash

VER=$(curl -sI https://github.com/ngoduykhanh/wireguard-ui/releases/latest | grep "location:" | cut -d "/" -f8 | tr
-d '\r')

echo "downloading wireguard-ui \${VER}"
curl -sL "https://github.com/ngoduykhanh/wireguard-ui/releases/download/\${VER}/wireguard-ui-\${VER}-linux-
amd64.tar.gz" -o wireguard-ui-\${VER}-linux-amd64.tar.gz

echo -n "extracting "; tar xvf wireguard-ui-\${VER}-linux-amd64.tar.gz

echo "restarting wgui-web.service"
```

```
systemctl restart wgui-web.service
```

```
EOF
```

```
chmod +x /etc/wireguard/update.sh
```

```
cd /etc/wireguard; ./update.sh
```

## Änderungen der Konfiguration verarbeiten

Mit dem folgenden Skript wird die **WireGuard Schnittstelle wg0** beobachtet. Bei Veränderungen an der Konfigurationsdatei wird dann **WireGuard** neu gestartet.

```
cat <<EOF > /etc/systemd/system/wgui.service
```

```
[Unit]
```

```
Description=Restart WireGuard
```

```
After=network.target
```

```
[Service]
```

```
Type=oneshot
```

```
ExecStart=/bin/systemctl restart wg-quick@wg0.service
```

```
[Install]
```

```
RequiredBy=wgui.path
```

```
EOF
```

```
cat <<EOF > /etc/systemd/system/wgui.path
```

```
[Unit]
```

```
Description=Watch /etc/wireguard/wg0.conf for changes
```

```
[Path]
```

```
PathModified=/etc/wireguard/wg0.conf
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
EOF
```

## WireGuard und WireGuard UI starten

Jetzt starten wir **WireGuard** und **WireGuard UI**.

```
touch /etc/wireguard/wg0.conf
systemctl enable wgui.{path,service} wg-quick@wg0.service wgui-web.service
systemctl start wgui.{path,service}
```

Wir können jetzt das **Web Interface** von **WireGuard UI** unter der **Public IP** und **Port 5000** öffnen. Die Installation ist damit abgeschlossen!

